Leveraging Renewable Energy in Edge Clouds for Data Stream Analysis in IoT

Yunbo Li, Anne-Cécile Orgerie, Ivan Rodero, Manish Parashar, Jean-Marc Menaud

Leveraging Renewable Energy in Edge Clouds for Data Stream Analysis in IoT

Yunbo Li*, Anne-Cécile Orgerie†, Ivan Rodero‡, Manish Parashar‡ and Jean-Marc Menaud*
*Mines Nantes, LINA, Nantes, France – Email: {yunbo.li, menaud}@mines-nantes.fr
†CNRS, IRISA, Rennes, France – Email: anne-cecile.orgerie@irisa.fr
‡RDI², Rutgers University, Piscataway, NJ, USA – Email: {irodero, parashar}@rutgers.edu

***Abstract*—The emergence of Internet of Things (IoT) is participating to the increase of data- and energy-hungry applications. As connected devices do not yet offer enough capabilities for sustaining these applications, users perform computation offloading to the cloud. To avoid network bottlenecks and reduce the costs associated to data movement, edge cloud solutions have started being deployed, thus improving the Quality of Service. In this paper, we advocate for leveraging on-site renewable energy production in the different edge cloud nodes to green IoT systems while offering improved QoS compared to core cloud solutions. We propose an analytic model to decide whether to offload computation from the objects to the edge or to the core Cloud, depending on the renewable energy availability and the desired application QoS. This model is validated on our application use-case that deals with video stream analysis from vehicle cameras.**

I. Introduction

The development of IoT (Internet of Things) community, the popularization of mobile devices, and emerging wearable devices brings new opportunities for context-aware applications in cloud computing environments [1]. Since 2008, the U.S. National Intelligence Council lists the IoT among the six technologies that are most likely to impact U.S. national power by 2025 [2]. The disruptive potential impact of IoT relies on its pervasiveness: it should constitute an integrated heterogeneous system connecting an unprecedented number of physical objects to the Internet [1]. A basic example of such objects includes vehicles and their numerous sensors.

Among the many challenges raised by IoT, one is currently getting particular attention: making computing resources easily accessible from the connected objects to process the huge amount of data streaming out of them. Cloud computing has been historically used as enable for a wide number of applications. It can naturally offer distributed sensory data collection, global resource and data sharing, remote and real-time data access, elastic resource provisioning and scaling, and pay-as-you-go pricing models [3].

However, it requires the extension of the classical centralized cloud computing architecture towards a more distributed architecture that includes computing and storage nodes installed close to users and physical systems [4]. Such an edge cloud architecture needs to deal with flexibility, scalability and data privacy issues to allow for efficient computational offloading services [5].

While computation offloading to the edge can be beneficial from a Quality of Service (QoS) point of view, from an energy perspective, it is relying on less energy-efficient resources than centralized Cloud data centers [6]. On the other hand, with the increasing number of applications moving on to the cloud, it may become untenable to meet the increasing energy demands which is already reaching worrying levels [7]. Edge nodes could help to alleviate slightly this energy consumption as they could offload data centers from their overwhelming power load [6] and reduce data movement. In particular, as edge cloud infrastructures are smaller in size than centralized data center, they can make a better use of renewable energy [8].

In this paper, we propose to leverage on-site renewable energy production in the different edge cloud nodes to green IoT. Our aim is to evaluate, on a concrete use-case, the benefits of edge computing regarding renewable energy consumption. We propose an analytic model for deciding whether to offload computation from the objects to the edge or to the core Cloud, depending on the renewable energy availability and the desired application QoS, in particular trading-off between performance (response time) and reliability (service accuracy). Our validation use-case targets the Internet of Vehicles (IoV) which can be seen as a convergence of the mobile internet and the IoT [9]. In particular, we focus on video streams from cameras that need to be analyzed usually for object detection and tracking. In this particular case, as it is often the case with IoT applications, a high QoS level is required. Indeed, data lose their value when they cannot be analyzed fast enough.

II. Related Work

A. Offloading Data to edge

Processing the data streams analysis consumes enormous computational resources and the response time is usually crucial for many applications. Moving the data to the cloud for analysis can be a solution [10] in a variety of application scenarios that require enormous computational resources as well as QoS guarantees. However, it might pose a risk of network bottleneck that thousands data streams are produced from IoT devices at the same time and then transmitted to central cloud (**core**) for quick analysis. Although lowering the analysis time profits large computational resources from cloud, it cannot avoid the time for data transferring through

the network from user to the physical location of cloud which might be thousands miles [11]. Furthermore, the increasing number of data streams over the network consume a large amount of energy.

To meet the demand of low latency response times, computation offloading to edge can be a answer [12]. The edge represents small-scale data centers that are close to the data source. The concept of processing data at the edge is based on the advantage of lower latency than core, therefore been able to quickly return result to the device. Nevertheless, considering the large amount of data streams that need to be processed, the core which has more computational resources may be a good choice.

B. Renewable energy and energy storage devices

Besides, renewable energy in the world has grown strongly in recent years. One reason is the solar-power generation efficiency significant increase. It enables the small-/medium-scale data centers to generate their own renewable energy. Thus they become self-sustainable and allow to reduce the fossil fuels (brown energy) consumption. As a consequence of the renewable energy success, the cost of producing green energy is becoming cheaper than brown energy. The direct result is that the cost for the user to use the cloud to accomplish their tasks in this kind of data centers is falling in a similar way when renewable energy is available.

Unlike traditional infrastructures where energy sources are controllable, integrating renewable energy into a data center becomes difficult due to its intermittent and variable nature. Solar energy is considered as an admissible renewable source as solar panels are easy to install, they present a reasonable efficiency and the variations in their electricity production are not too abrupt (as for wind) [8]. Usually, most electricity generated by solar panels is during the day and its peak power always near the midday. However, workloads do not necessarily follow the renewable energy production which may result in a waste of energy. In order to increase the usage of renewable energy, one way consists in carefully scheduling the workload to align with the time-varying renewable energy. In [13], *Li et al.* propose an online algorithm making use of opportunistic scheduling for optimizing solar energy utilization in a small-/medium-scale data center without energy storage. This approach leverages two ideas: 1) delay part of jobs which could be suspended within limit time (e.g., batch jobs) until solar energy becomes available; 2) when the renewable energy production cannot fully support the entire workload energy consumption, the system migrate the jobs from under-utilized servers to others and switches-off them with the help of consolidation techniques. However, this approach offers an efficient solution where the jobs are delay-tolerant (e.g., batch jobs). In this contribution, we allow some jobs to be delayed in order for the workload to follow the renewable energy generation that maximizes the green energy usage.

In [13], *Li et al.* explain that such a system cannot be satisfied when the workload contains real-time jobs. The proposed solution consists in using Energy Storage Devices (ESDs) [14] to store the surplus electricity generated from renewable energy sources. By integrating ESDs, real-time jobs always have access to green energy and so they are not forced to be delayed. Nevertheless, a penalty is occurred that storing the renewable energy into batteries leads a energy loss. Yet, storing renewable energy into batteries leads to an energy loss because of energy transformation. In particular, the renewable energy in this paper refers solar energy.

C. Video streams analysis

Existing literature has addressed video analysis algorithms and tools. Haar feature-based cascade classifiers [15] is a typical method for object detection which is effective and capable of achieving high detection rates. It is based on machine learning approach AdaBoost [16] and trains a cascade function from a large set of positive and negative images. The classifiers are included in the OpenCV distribution 2.4.13, we trained our own a Haar classifier which is used to analyze video streams for vehicles detection in this paper.

III. Driving Use Case

A. Continuous data streaming in edge computing systems

The edge typically has less computing capacity (e.g., compute servers) than the resources available in the cloud core. However, theses edge servers are closer to the edge-users and therefore the latency to edge servers is lower than the latency to the core. We consider that edge servers have dual energy supply which include traditional brown energy and renewable energy with a reasonably sized Energy Storage Device (ESD) to store the surplus renewable energy.

The core represents the federation of large data centers where each data center is composed of thousands of servers. Such a federation model of data centers [17] with federation of resources and autonomic management mechanism offers a large pool of computing resources. While the core has more powerful servers the energy costs associated to data movement present different tradeoffs that need to be investigated.

The motivation of this work is to provide a framework that can balance performance and energy cost tradeoffs for real-time data analysis of high-rate data from many sensors. A typical use case scenario is the camera, which can be embedded in small devices as such Google Glass, GigaSight [18] or any other devices. The camera captures frames continuously that can be seen as a high-rate data stream. Since such a video analysis that detects interesting objects (i.e., areas of interest) from it, the analysis will consume most of computation resources and thus energy. To increase the computation performance and reduce energy consumption on the end device, data is often offloaded to the Cloud to be analyzed. Although data offloading to

high performance servers at the Cloud can accelerate the analysis processing, the efficiency of the whole procedure is highly dependent to the network condition and to the costs associated to the network service.

In this paper, we make the assumption that all the vehicles are equipped with an on-board camera and are capable for uploading the video captured by their cameras continuously to edge and core clouds. The edge/core analyzes each data stream in real time and return the road condition to the user. The goal is preventing traffic jam and possible traffic accidents by sharing the produced information to users in an online manner. Integrating this into next generation of vehicles with autopilot technology can help improving the road safety for the drivers (i.e., the users). As shown in Figure 1, an object is detected by analyzing the data stream from the first car, the resulting analysis identifies an object in the middle of the road which may be dangerous for the other vehicles behind on this road. The edge-1 immediately informs all the vehicles that are in section BC of the road. At the same time, a message is sent from edge-1 to the edge-0 in order to inform the vehicles in section AB of the road.

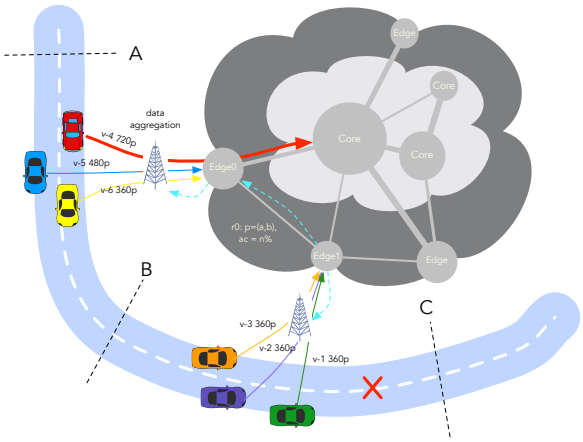


Figure 1: Use case for IoV

IV. System model and assumptions

A. Edge and Core model

Inspired by the previous work on video stream analysis [18], [19] and edge-computing [20], our model involves 2 types of computing resources.

1) Computation at edge: due to user is physically close to the edge, the servers place at the the edge enables low latency for users. The data transfers from user to edge can have a lower latency than direct transferring to the Core Cloud. Conversely, the computation capacities at the Edge cloud is limited and can be seen as a small-scale data center, the considered edge comprises between 20 to 50 servers. Each server has limited physical resources in terms of CPU, RAM and ingress bandwidth. We assume that there is no centralized storage system at the edge cloud: each server has its own hard disk [21]. Once the edge cannot satisfy the computational task QoS requirement, it transfers the task to core where sufficient computing resources are available.

The edge is equipped with a number of photovoltaic (PV) panels and an ESD. It has dual brown (from regular grid) and renewable energy supplies. If the renewable energy cannot be entirely consumed by edge servers, the ESD stores the surplus of renewable energy for future use. We also assume that each server has a switch connected with renewable, brown energy supplies and the ESD. In particularly, the server can only opt for using one of the three sources at the same time.

2) Computation at core: the core represents a federation of inter-connected data centers which are usually far from users. Although the servers place at the core cloud have higher latency than edge servers, whether the number of servers or the performance of core server that are higher than edge. From the energy cost perspective, the data processing at the core is faster than data processing at the edge. However, a large volume of data need to be transferred to core to process that the communication cost between user-core through the Internet cannot be ignored.

A job is a request from a vehicle that requires computational resources for processing. It can be submitted to the edge and the core at anytime. Once the request is accepted, a Virtual Machine (VM) is created on a server at the edge or core to process the analysis. A VM is considered as the basic unit of resource allocation. Each VM is created with its specific vCPU and RAM. When the vehicle leaves this section of road, the VM is destroyed and it releases its reserved resources back to the server.

B. Renewable energy and ESD model

Due to the variable and intermittent nature of solar energy, an energy production prediction is performed while a job scheduling decision has to be taken. It predicts only the amount of solar energy for the next time slot (1 hour), so that such short-time prediction is able to achieve a high accuracy [8]. To simplify the problem, we assume that the prediction error ratio approaches 0 in our validation methodology.

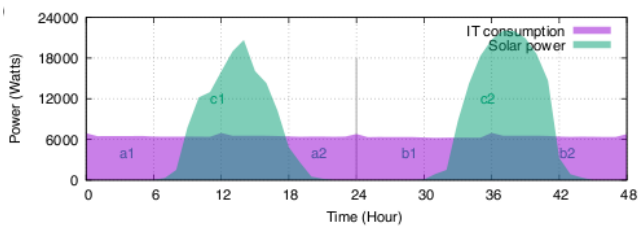


Figure 2: ESD

As shown in Figure 2, the purple curve $w(t)$ represents the workload energy consumption and the green curve $g(t)$

represents the solar power. We observe that for areas a_1, a_2 and b_1, b_2, the solar energy supply is lower than workload energy consumption. Without an ESD, the total energy consumption from the grid can be expressed as:

$$E_{\text{brown}} = \sum_{t=0}^{t_1} (w(t) - g(t)), \forall (w(t) > g(t)), t \in T \quad (1)$$

When the solar energy is higher than workload energy demand, the amount of surplus solar energy is defined as:

$$E_{\text{surplus}} = \sum_{t=t_1}^{t_2} (g(t) - w(t)), \forall (w(t) < g(t)), t \in T \quad (2)$$

For day 1 on Figure 2, $E_{brown} = a_1 + a_2$ and $E_{surplus} = c_1$. The ESD can be charged when $w(t) < g(t)$. When the solar energy is not sufficient to supply for the current workload energy, we first discharge the battery. Once the ESD runs out, the servers then consume the brown energy from the grid.

The capacity of the ESD is finite. Herein, we define the maximum capacity C of an ESD. The energy that has been collected at a given time represents as t, $C_{available}(t)$ and is stored by the ESD. In order to extend the battery lifetime, we take into account the Depth-of-Discharge (DoD) constraint [22], [23], which stipulates that the remaining energy stored in an ESD has to be larger than the DoD threshold. So, in other terms, the available stored energy is lower than a higher bound ηC ($0 < \eta < 1$, e.g, $\eta = 0.8$). Due to the DoD constraint, the $C_{available}(t)$ never reaches C. Formally, we have $0 \leq C_{available}(t) \leq \eta C$.

An ESD has two significant functionalities: charging (collects energy from solar panels) and discharging (powers the data center). In our model, we consider that charging and discharging are two independent procedures. It implies ESD is never under charging and discharging states simultaneously. The charging rate is limited by an upper bound λ that mainly depending on the ESD type and capacity. During a given time period $[t_i, t_j]$ ($t_j > t_i$), if we suppose the available renewable energy (supplied by PV cells) is $E(t_i, t_j)$, we employ formula 3 to compute the amount of energy $E_{in}(t_i, t_j)$ can be stored into an ESD.

$$E_{\text{in}}(t_i, t_j) = \min(E(t_i, t_j), \lambda(t_j - t_i), C_{\text{available}}(t_i)) \times \sigma \quad (3)$$

Parameter σ is a constant that describes the energy efficiency of the battery's charging procedure. The discharging rate is also limited by an upper bound denoted μ. During a consecutive time period $[t_i, t_j]$, we use formula 4 to compute the amount of energy $E_{out}(t_i, t_j)$ provided by the ESD. Parameter $E_{\text{self-discharge}}(t_j - t_i)$ represents the energy loss because of the self-discharging of batteries.

$$E_{\text{out}}(t_i, t_j) = \min(\mu(t_j - t_i), \eta C - C_{\text{available}}) - E_{\text{self-discharge}}(t_j - t_i) \quad (4)$$

V. Experimentation

The first half of our experiment is to measure the power consumption and performance degradation with different resolutions on Grid'5000, a French platform for experimenting distributed system [24]. The used servers are Dell PowerEdge R720 from the *Taurus* cluster at Grid'5000 Lyon site. Each server is composed of two Intel Xeon E5-2630 processors (2.3GHz) each with 6 cores, 32 GB of RAM and 600 GB of disk space. The processors support hyper-threading technology thus the total of 12 physical cores servers can provide 24 virtual CPUs. KVM is the virtualization solution along with Linux on x86-based servers. The experiment results are used for building power and performance models. The network energy consumption model is defined in a similar way in [25] and based on bit. These models were integrated into the simulator we developed in [13]. In order to extrapolate to large-scale, the second half of our experiments are held using this simulator.

A. Setup

The servers are placed at both edge and core. The server power consumption is related to different components. Most of previous studies [26] agree on the fact that the dynamic server power consumption mainly depends on the working CPU frequency. The server power consumption is taken for different CPU load profiles as described in [13]. Furthermore, our experimental results show in particular that a server on idle state consumes roughly half of its maximal power consumption. From the latency point-of-view, we assume a 100 ms Round-Trip-Time (RTT) between the vehicles and the core cloud. This value is similar to what can be observed for accessing an Amazon Cloud for instance [27].

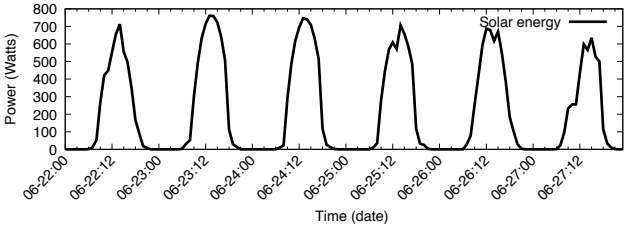
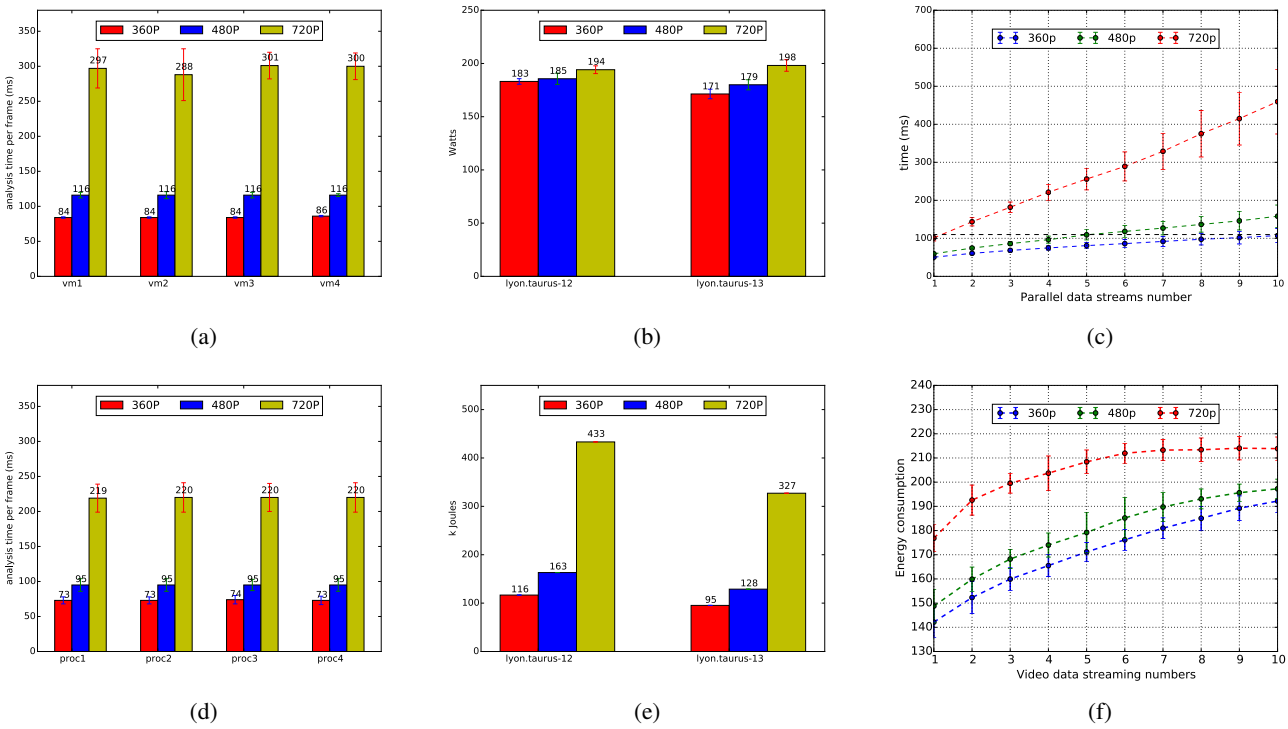


Figure 3: Solar energy production with solar panels of 5.52 m^2

On the side of the solar power production, we employ a mini-scale solar power farm which is set up in the campus of University Nantes[1]. It is composed by 8 identical panels *Sanyo HIP-240-HDE4* and *SMA Sunny Boy 1200* inverter. The theoretical max power of each panel is 240 Watt. Subsequently, we extract a whole week data (22-28 June 2015) from the database which is shown in Figure 3, the days in this week are mostly sunny.

[1]Traces available online: http://photovolta2.univ-nantes.fr



(a) (b) (c)

(d) (e) (f)

Figure 4: Energy consumption and frame analysis time of resolution in 360p, 480p and 720p

B. *VM size and time analysis*

Due to the server limited computational capacity, allocating resources to VMs needs to be carefully done. The goal of our first experiment is to evaluate the video analysis performance and energy consumption on different size of VMs. In this experiment, we create two individual VMs on two servers from the Taurus cluster. The VM-1 is given 2 vCPU and 2 Gb RAM, and the VM-2 is given 4 vCPU and 4Gb RAM.

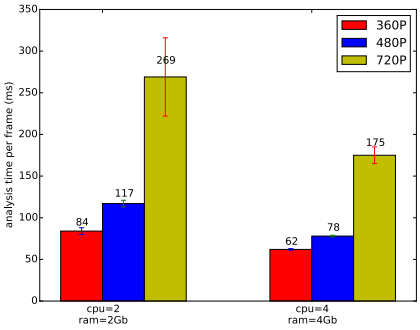


Figure 5: Time analysis on different VM sizes

The time analysis per frame of VM-1 and VM-2 are shown in Figure 5. VM-2 is 26%, 33% and 35% faster than VM-1 in resolution of 360p, 480p and 720p respectively. Clearly, the VM-2 benefits from more computational resources and it results in a reduced analysis time.

We then move on to another experiment where we variate the VMs size. We first create VM-1∼4 on server Taurus-12, each VM has the same hardware configuration: 2 vCPU and 2 Gb RAM. Theses VMs process only 1 data stream at a time. VM-5 is created on server Taurus-13 with 8 vCPU and 8 Gb RAM. Unlike VM-1∼4, it processes 4 data streams in parallel. We conducted the experiments on analyzing the same video. The video is encoded through H.264 codec in 3 resolutions (360p, 480p and 720p) and we use the FFmpeg tool [28] for decoding. The experiment iterates 10 times and each time only processes 1 format of video.

The results are shown in Figure 4. Figure 4a is when 4 individual small VMs are used and each VM only processes 1 data stream. In Figure 4d, it shows the processing of 4 data streams in parallel within a large size VM. We observe that processing 4 streams in 1 large VM is faster than processing in 4 small size VMs. We attribute this to the fact that the KVM virtualization layer adds a penalty. In case of 4 VMs, the computational resources given to each VM from KVM is not always from the same physical cores. In other words, there is a scheduling cost if a VM is not always using at least one physical core. As shown in Figure 4b, the average power consumption (on Watt) for processing 4 data streams in 1 larger VM is lower compared with 4 small VMs. For analyzing a 5 minute video, as shown in Figure 4e, VM-5

with faster speed of frame analysis and lower instantaneous power consumption, it consumes less energy in total.

We also observe that the processing time increases significantly with the resolution increasing. For each video stream, we expect to analyze 8 frames per second. It means that we have to analyze 1 frame in every 3 frames with a video at 25 fps (i.e., the average analysis time per frame must be inferior than 125 ms). To compute the maximum number of videos that can be analyzed in parallel, we assume that 1 VM is used for analyzing 1 format of videos. We measure the time analysis on VM-5 for a video in 3 resolutions. As shown in Figure 4c, VM-5 supports in parallel up to 11 videos streams in resolution 360p, 4 video streams for 480p video and only 1 for 720p video. Figure 4f shows the respective energy consumption in the 3 resolutions for VM-5.

C. Edge-/core-energy consumption

In this subsection, we evaluate the effect of offloading computation tasks at the edge for system performance of our framework and energy consumption at edge and core. We study the scalability of our framework by increasing the number of vehicles (source videos). We assume that there is no bottlenecks in the network between user-edge and edge-core. The experiments in this subsection are performed using simulation.

Edge usually has less computational resources in comparison with core. In initial configuration, edge has 5 servers and dual energy consumption (self-produced renewable energy with ESD and brown) and core has 100 servers without any renewable energy source. Each edge server has 24 vCPU and 24 Gb and the core servers are twice as powerful as edge servers. To avoid the energy consumption associated with VM placement, we assume all the VMs are same size that consists of 8 vCPU and 8 Gb RAM at edge. The VMs have 24 vCPU and 24Gb at core implying the time analysis is reduced. We only consider 360p and 720p videos in this scenario in order to illustrate that different resolutions that impacts energy consumption and performance. As mentioned before (section V-B), a VM processes one format of video in the experiment thus a VM can maximum process 1 video stream for 720p, and 10 video streams for 360p in parallel as shown in Figure 4c. All the requests of data analysis are processing at the edge by default. If edge doesn't have sufficient resources for processing, these request they will be transferred to core.

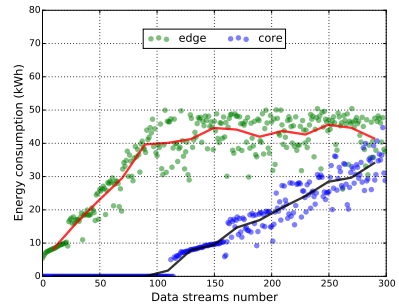
The goal of this experiment is to measure the total energy consumption at both the edge and core. We first assume that all the data streams are 360p and the renewable energy is not available at the edge (e.g.,there is no solar energy production during the night). At beginning, there are few vehicles in the system. These vehicles first offload their data to edge to process. With increasing the number of data streams, the edge energy consumption increases by processing these data streams. As shown in Figure 6a, we observe that the core doesn't consume any energy before the edge computational resources are exhausted. It starts to process data when the number of data streams exceeds 112 in the system. In Figure 6b, all the 360p videos are replaced by 720p, the edge quickly drained its resources where processing 720p videos consumes more computational resources than 360p videos. The core receives the first request of data analysis from the 16th vehicle. From that moment, all the new data arrivals are directed to core to process.

Once the renewable energy becomes available, as shown in Figure 6d, the edge consumes directly the renewable energy (green) instead of brown energy (gray). The surplus renewable energy produced is stored into its ESD for future usage. Edge is always prior to consume from renewable energy source and then consume from its ESD. It consumes the brown energy when the both are unavailable. Figure 6c shows, by integrating the renewable energy and ESD at edge, it reduces roughly half of energy consumption compared with non-renewable energy configuration.

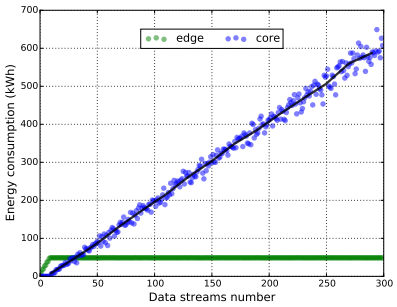
In Figure 6e, we can observe that the average delay of 360p videos is significantly lower than 720p videos. Because of almost analysis tasks are performed at edge instead of at core. Once the edge has exhausted all the resources, the new arrivals are migrated to perform at the core. On the scale of 300 vehicles, edge is capable of processing 37.3% 360p videos streams in the system. In contrast to 360p, processing 720p video stream consumes much more computational resources than processing 360p videos. The edge can only process 5% data streams and all the other data streams have to move to core for processing. Despite the core possesses more powerful computational resources which might even reduce the time analysis, the latency from the network between edge and core that cannot be ignored. Figure 6e also demonstrates that the average delay of all videos are mainly depending on the number of data streams offloading to the core. With increasing the data streams moving to core, the network energy consumption is also increased.

D. The detection accuracy and number of cameras

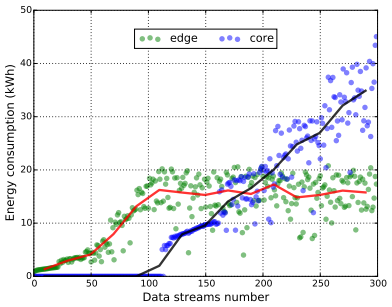
Processing analysis in higher resolution video often output a result with high detection accuracy. However, it consumes enormous computational resources including CPU/RAM and bandwidth for transmission. Reducing the resolution is a clear way to save computational resources and network utilization. Edge servers can process more videos streams in parallel without significant performance degradation. It potentially decreases network usage thus more video streams can be processed at edge. However, scaling down the video affects the detection accuracy. As mentioned in [18], lowering the resolution of video significantly reduces the detection accuracy. As shown the initial accuracy setting in this subsection for object detection in Table. I



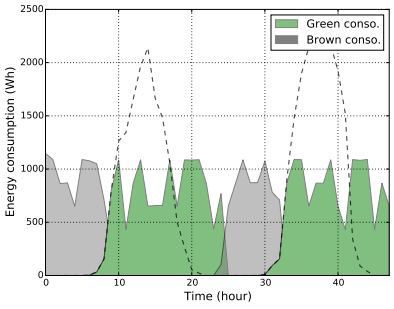
(a) Resolution 360p



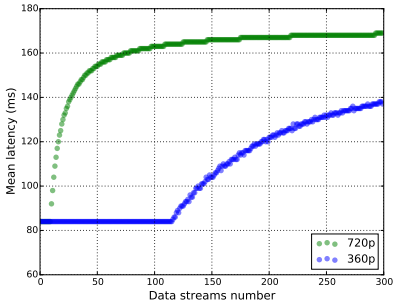
(b) Resolution 720p



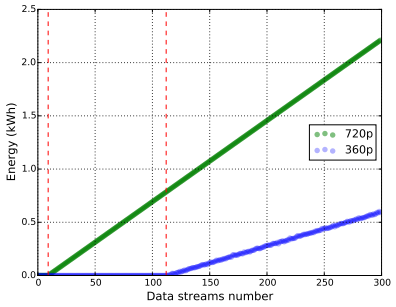
(c) Resolution 360p



(d) 2 days (48 hours) energy consumption at edge



(e) Average delay in the system



(f) Data offloading affects network energy consumption between edge and core

Figure 6: The renewable energy is not available at edge in Figure (a) and (b) and is available in Figure (c)

Classes	720p	480p	360p
car	96.7%	91%	88.5%
body	97.7%	94.9%	90.7%
dog	96.1%	94.9%	90.7%
total	96.7%	92.3%	87.9%

Table I: The detection accuracy of different objects [18]

Assuming that there is only one car in the section *AB* of road, the detection accuracy for car is equal to 96.7%, 91% , 88.5% (720p, 480p, 360p respectively). Now, we assume that there are two cars in the same section, their cameras both capture on resolution 360p that the accuracy of detection is same. When one of the two cameras detected a object on the road and another didn't. In this case, which one should be used for the definitive result? Furthermore, we replace one camera by using 720p resolution. Suppose the two results are still different, should us do always believe the result with higher resolution (720p) that because of its higher detection accuracy by default?

Unfortunately, we cannot directly conclude which result of the two is more believable. Even though the 720p videos often offers a higher detection accuracy than 360p videos, this only shows that 720p is more likely to be correct, but not conclusive. However, with increase in the number of cameras, we shows that the correct probability of result is not only depending on the initial detection accuracy, but also related to the number of cameras in the system. Suppose there are $2n+1$ cars in the same section of road. All the car upload video stream in same resolution and then they output $2n+1$ results. Intuitively, if there is a result appeared at least half of the total, we prefer to select this result as final result. We define the **reliability** is the probability of a result appears exceeds $n+1$ times among $2n+1$ results. In this section, we prove this final result becomes more believable when the number of cameras increase. First, each result is independent with others, so the reliability can be expressed as following:

$$\begin{aligned}\text{reliability} &= \Pr(X \geq n+1) \\ &= \sum_{x=n+1}^{2n+1} C_{2n+1}^{x} (p)^{x} (1-p)^{2n+1-x} \\ &= \sum_{x=n+1}^{2n+1} \binom{2n+1}{x} (p)^{x} (1-p)^{2n+1-x} \\ &= \sum_{x=n+1}^{2n+1} \frac{(2n+1)!}{(2n+1-x)!x!} (p)^{x} (1-p)^{2n+1-x} \\ &\text{where } p \in (0,1)\end{aligned} \tag{5}$$

After simplifying the equation 5 , it can be expressed:

$$\text{reliability} = \frac{1}{1+\left(\frac{1-p}{p}\right)^{2n+1}} = \frac{1}{1+\omega^{2n+1}}, \text{where } \omega = \frac{1-p}{p} \tag{6}$$

(More details on the simplification of equation 6 can be found in the Appendix page.)

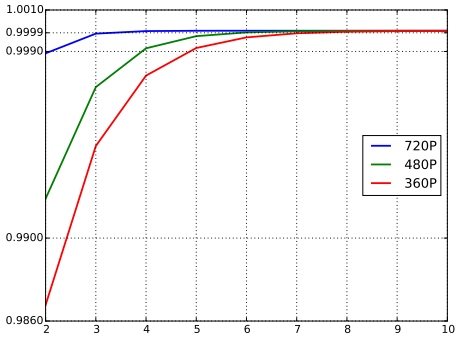


Figure 7: Reliability

From equation 6, When $p > 0.5$, ω decrease by increasing n and the reliability increases monotonically. While $n \to \infty$, the value of reliability is infinitely close to 1. In reverse, $p < 0.5$, ω increase by increasing n and the reliability decreases monotonically. While $n \to \infty$, the value of reliability is infinitely close to 0. When $p = 0.5$ and $n \to \infty$, the reliability is infinitely close to 0.5.

The significance of equation 6 is, more than half of total results all point to a same result, the correct probability of this result is approaching to 100%. It also shows this probability is cumulative by increasing the number of video process. Although we cannot change the initial detection accuracy for each format, we are still able to reduce the probability of returning an erroneous result. e.g., it is able to reach up to 99.999% or even higher probability for assuring the result is on the side of 96.7% instead of 3.3%. In other words, the occurrence of this 3.3% can approach infinitely to 0% while the number of video process is increasing.

We introduce the **nines** conception which is typically expressed as a percentage with a number of nines. e.g., $99\% \to$ two nines, $99.9\% \to$ three nines etc. This conception is similar with the conception of *High availability* in system design which aims to ensure an agreed level of operational performance.

# **nines**	720p	480p	360p
99.9%	3	4	6
99.99%	4	6	8
99.999%	5	7	11

Table II: The number of cameras needed for achieving the indicate number of nines.

As shown in Figure 7 and Table II, the resolution 360p requires 6 cameras working on simultaneously that can achieve *three nines*, the 480p requires 4 cameras and the 720p requires only 3 cameras to achieve the same level. The higher resolution is, the less number of camera are required for reaching a same level of reliability.

VI. Discussion

As mentioned in previous sectionV-C, the edge is capable of generating its own energy and storing the surplus energy into an ESD, the result shows that the renewable energy almost covers its total energy consumption. Due to its limited computational resources, it cannot support while amount of process that needs to occur at the same time. All the incoming data streams have to move to core cloud for quick analysis. As well as we conclude previously, to reduce the brown energy consumption, it is better to reduce the resolution for all videos with a penalty on detection accuracy. From an environmental point of view , if the user expects high accuracy of detection and to consume clean energy instead of brown, it first needs to ensure that the data is processing at the edge. As the number of user grows, we then have to increase not only the number of edge servers, but also the solar voltaic panels that are able to provide as the same amount of energy as the servers need.

To reduce the total brown energy consumption, another alternative solution is changing the division of labor for edge. The finite computational resources at edge is no longer used for data analysis the data but for decoding, sampling and encoding. As such 720p videos consumes particularly a lot of computational resources, even taking all the edge servers, it is still far from enough for processing all the 720p videos in the system. Thus, careful using edge resources is important for the framework. As described in section V, it needs to analyze 8 frames every second for a video at 25 FPS. It means that we select 1 frame out every 3 frames for processing. In particularly, we expect the sampling work can be done at the edge. When a new video stream arrives, the edge performs decoding, sampling and encoding successively on this video and then transfers it to core. Although the data has to move to core for processing, it reduce its size and the energy consumption over the network is also reduced. Unfortunately, the result of this experiment is unsatisfactory. Decoding a video at 720p is extremely fast but encoding will take 15x times than decoding in our experiment. It leads an additional delay roughly 100 ms where the latency is crucial in this scenario.

VII. Conclusion

Data looses its value when it cannot be analyzed quick enough. Offloading the data to process analysis at edge significantly reduces the response time and avoid unnecessary data transmission between edge-core. Building self-producing electricity edge can further reduce the traditional energy consumption and carbon footprint of these energy-hungry infrastructures. Although our works are camera-based, it can be applied to any other scenarios where the

data streams need to be processed in real-time as it provides the analytic framework for such applications.

ACKNOWLEDGMENT

This work has received a French state support granted to the CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference Nb. ANR-10-LABX-07-01. The research presented in this work is supported in part by National Science Foundation (NSF) via grants numbers ACI-1464317, ACI-1339036, ACI-1310283, and CNS-1305375. We thank Yifu Tang for numerous fruitful discussions.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] S. C. B. Intelligence, "Six Technologies with Potential Impacts on US Interests out to 2025," National Intelligent Concil, Tech. Rep., 2008.

[3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, "Enabling Smart Cloud Services Through Remote Sensing: An Internet of Everything Enabler," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 276–288, June 2014.

[4] L. M. Vaquero and L. Rodero-Merino, "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[5] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan, "Quantifying the Impact of Edge Computing on Mobile Applications," in *ACM SIGOPS Asia-Pacific Workshop on Systems*, 2016, pp. 5:1–5:8.

[6] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and Opportunities in Edge Computing," *CoRR*, vol. abs/1609.01967, 2016.

[7] T. Bawden, "Global warming: Data centres to consume three times as much energy in next decade, experts warn," Independent, http://www.independent.co.uk/environment/global-warming-data-centres- to-consume-three-times-as-much-energy-in-next-decade-experts-warn- a6830086.html, 2016.

[8] Í. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: managing datacenters powered by renewable energy," in *ACM SIGARCH Computer Architecture News*, vol. 41, no. 1. ACM, 2013, pp. 51–64.

[9] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, 2014.

[10] A. Ishii and T. Suzumura, "Elastic stream computing with clouds," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 195–202.

[11] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54–61, 2016.

[12] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.

[13] Y. Li, A.-C. Orgerie, and J.-M. Menaud, "Opportunistic scheduling in clouds partially powered by green energy," in *GreenCom: IEEE International Conference on Green Computing and Communications*, 2015, pp. 448–455.

[14] Y. Ghiassi-Farrokhfal, S. Keshav, and C. Rosenberg, "Toward a realistic performance analysis of storage systems in smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 402–410, 2015.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001.

[16] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[17] J. Diaz-Montes, M. Zou, I. Rodero, and M. Parashar, "Enabling autonomic computing on federated advanced cyberinfrastructures," in *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*. ACM, 2013, p. 20.

[18] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable crowd-sourcing of video from mobile devices," in *ACM International conference on Mobile systems, applications, and services*, 2013, pp. 139–152.

[19] A. Anjum, T. Abdullah, M. Tariq, Y. Baltaci, and N. Antonopoulos, "Video stream analysis in clouds: An object detection and classification framework for high performance video analytics," *IEEE Transactions on Cloud Computing*, 2016.

[20] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *arXiv preprint arXiv:1605.05488*, 2016.

[21] N. Beldiceanu, B. Dumas Feris, P. Gravey, S. Hasan, C. Jard, T. Ledoux, Y. Li, D. Lime, G. Madi-Wamba, J.-M. Menaud, P. Morel, M. Morvan, M.-L. Moulinard, A.-C. Orgerie, J.-L. Pazat, O. H. Roux, and A. Sharaiha, "Towards energy-proportional Clouds partially powered by renewable energy," *Computing*, p. 20, 2016.

[22] H. Chen, T. N. Cong, W. Yang, C. Tan, Y. Li, and Y. Ding, "Progress in electrical energy storage system: A critical review," *Progress in Natural Science*, vol. 19, no. 3, pp. 291–312, 2009.

[23] K. Divya and J. Østergaard, "Battery energy storage technology for power systems?an overview," *Electric Power Systems Research*, vol. 79, no. 4, pp. 511–520, 2009.

[24] D. Balouek *et al.*, "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud Computing and Services Science*. Springer, 2013, vol. 367, pp. 3–20.

[25] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, 2016.

[26] A.-C. Orgerie, M. Dias de Assunção, and L. Lefèvre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 47, 2014.

[27] I. Cuadrado-Cordero, F. Cuadrado, C. Phillips, A.-C. Orgerie, and C. Morin, "Microcities: a Platform based on Microclouds for Neighborhood Services," Inria, Research Report RR-8885, 2016.

[28] "https://www.ffmpeg.org," https://www.ffmpeg.org.

VIII. Appendix

Proposition: $\sum_{x=n+1}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x}$ increase monotonically by increasing n while $p > 0.5$. **i.e.,**

$$\sum_{x=n+1}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x} < \sum_{x=n+1}^{2n+2} C_{2n+2}^{x}(p)^{x}(1-p)^{2n+2-x} \tag{7}$$

proof:

$$\begin{aligned}
\Pr(X \geq 0) &= \sum_{x=0}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x} \\
&= C_{2n+1}^{0}[\underbrace{(p)^{0}(1-p)^{2n+1}}_{a_0} + \underbrace{(p)^{2n+1}(1-p)^{0}}_{b_0}] \\
&+ C_{2n+1}^{1}[(p)^{1}(1-p)^{2n} + (p)^{2n}(1-p)^{1}] \\
&+ C_{2n+1}^{2}[(p)^{2}(1-p)^{2n-1} + (p)^{2n-1}(1-p)^{2}] \\
&+ ... \\
&+ C_{2n+1}^{n}[\underbrace{(p)^{n}(1-p)^{n+1}}_{a_n} + \underbrace{(p)^{n+1}(1-p)^{n}}_{b_n}]
\end{aligned} \tag{8}$$

Where $a_0 = C_{2n+1}^{0}(p)^{0}(1-p)^{2n+1} = (1-p)^{2n+1}$. Then we simplify the common ratio $\frac{a_n}{a_{n-1}}$ and separate two sub-sequences a_n and b_n from $\Pr(X \geq 0)$

$$\begin{aligned}
\frac{a_n}{a_{n-1}} &= \frac{\frac{(2n+1)!}{(n+1)!n!}(p)^{n}(1-p)^{n+1}}{\frac{(2n)!}{(n-1)!n!}(p)^{n-1}(1-p)^{n}} \\
&= \frac{(1-p)}{p} \times \frac{(2n+1)n}{(n+1)n} \\
&= \frac{2(1-p)}{p} \times \frac{2n+1}{n}
\end{aligned} \tag{9}$$

The sub-sequence a_n can be expressed as:

$$\begin{aligned}
a_n &= C_{2n+1}^{n}(p)^{n}(1-p)^{n+1} \\
&= \frac{(2n+1)!}{(n+1)!n!}(p)^{n}(1-p)^{n+1} \\
&= a_0 \cdot \left(\frac{2(1-p)}{p}\right)^{n} \cdot \frac{(2n+1)(2n-1)(2n-3)\cdot ... \cdot 5 \cdot 3 \cdot 1}{(n+1)!}
\end{aligned} \tag{10}$$

Similarly, the sub-sequence b_n can be expressed as:

$$b_n = b_0 \cdot \left(\frac{2(1-p)}{p}\right)^{n} \cdot \frac{(2n+1)(2n-1)(2n-3)\cdot ... \cdot 5 \cdot 3 \cdot 1}{(n+1)!} \tag{11}$$

where $b_0 = p^{2n+1} \cdot (1-p)^{0} = p^{2n+1}$. As we know,

$$(a+b)^{n} = \sum_{x=0}^{n} C_{n}^{x}(a)^{x}(b)^{n-x} \tag{12}$$

So,

$$\begin{aligned}
(a+b)^{(2n+1)} &= \sum_{x=0}^{2n+1} C_{2n+1}^{x}(a)^{x}(b)^{2n+1-x} \\
(p+(1-p))^{(2n+1)} &= \sum_{x=0}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x} \\
1^{(2n+1)} &= \sum_{x=0}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x}
\end{aligned} \tag{13}$$

Let

$$h_n = \left(\frac{2(1-p)}{p}\right)^{n} \cdot \frac{(2n+1)(2n-1)(2n-3)... \cdot 5 \cdot 3 \cdot 1}{(n+1)!} \tag{14}$$

Accordingly, the sum of sub-sequennce a_n can be transformed as following:

$$\begin{aligned}
S_a &= \sum_{x=0}^{n} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x} \\
S_a &= a_0 + a_1 + a_2 + a_3 + ... + a_n \\
&= a_0 + h_1 \cdot a_0 + h_2 \cdot a_0 + h_3 \cdot a_0 + ... + h_n \cdot a_0 \\
&= a_0 \cdot (1 + h_1 + h_2 + h_3 + ... + h_n)
\end{aligned} \tag{15}$$

b_n is transformed in a similar way:

$$\begin{aligned}
S_b &= \sum_{x=n+1}^{2n+1} C_{2n+1}^{x}(p)^{x}(1-p)^{2n+1-x} \\
&= b_0 \cdot (1 + h_1 + h_2 + h_3 + ... + h_n)
\end{aligned} \tag{16}$$

The ration between S_a and S_b can be simplified:

$$\begin{aligned}
\frac{S_a}{S_b} &= \frac{a_0}{b_0} \\
&= \frac{p^{0} \cdot (1-p)^{2n+1}}{p^{2n+1} \cdot (1-p)^{0}} \\
&= \left(\frac{1-p}{p}\right)^{2n+1} \\
S_a &= \left(\frac{1-p}{p}\right)^{2n+1} \cdot S_b
\end{aligned} \tag{17}$$

Finally, the equation 13 can be transformed as:

$$\begin{aligned}
S_a + S_b = 1 &= \left[\left(\frac{1-p}{p}\right)^{2n+1} + 1\right] \cdot S_b \\
S_b &= \frac{1}{1 + \left(\frac{1-p}{p}\right)^{2n+1}}
\end{aligned} \tag{18}$$